(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 1 246 131 A2

(12) **EUROPEAN PATENT APPLICATION**

(43) Date of publication:
02.10.2002 Bulletin 2002/40

(51) Int Cl.7: **G06T 9/00**

(21) Application number: **02006295.6**

(22) Date of filing: **20.03.2002**

(84) Designated Contracting States:
AT BE CH CY DE DK ES FI FR GB GR IE IT LI LU MC NL PT SE TR
Designated Extension States:
AL LT LV MK RO SI

(30) Priority: **26.03.2001 US 817701**

(71) Applicant: **Sharp Kabushiki Kaisha**
Osaka (JP)

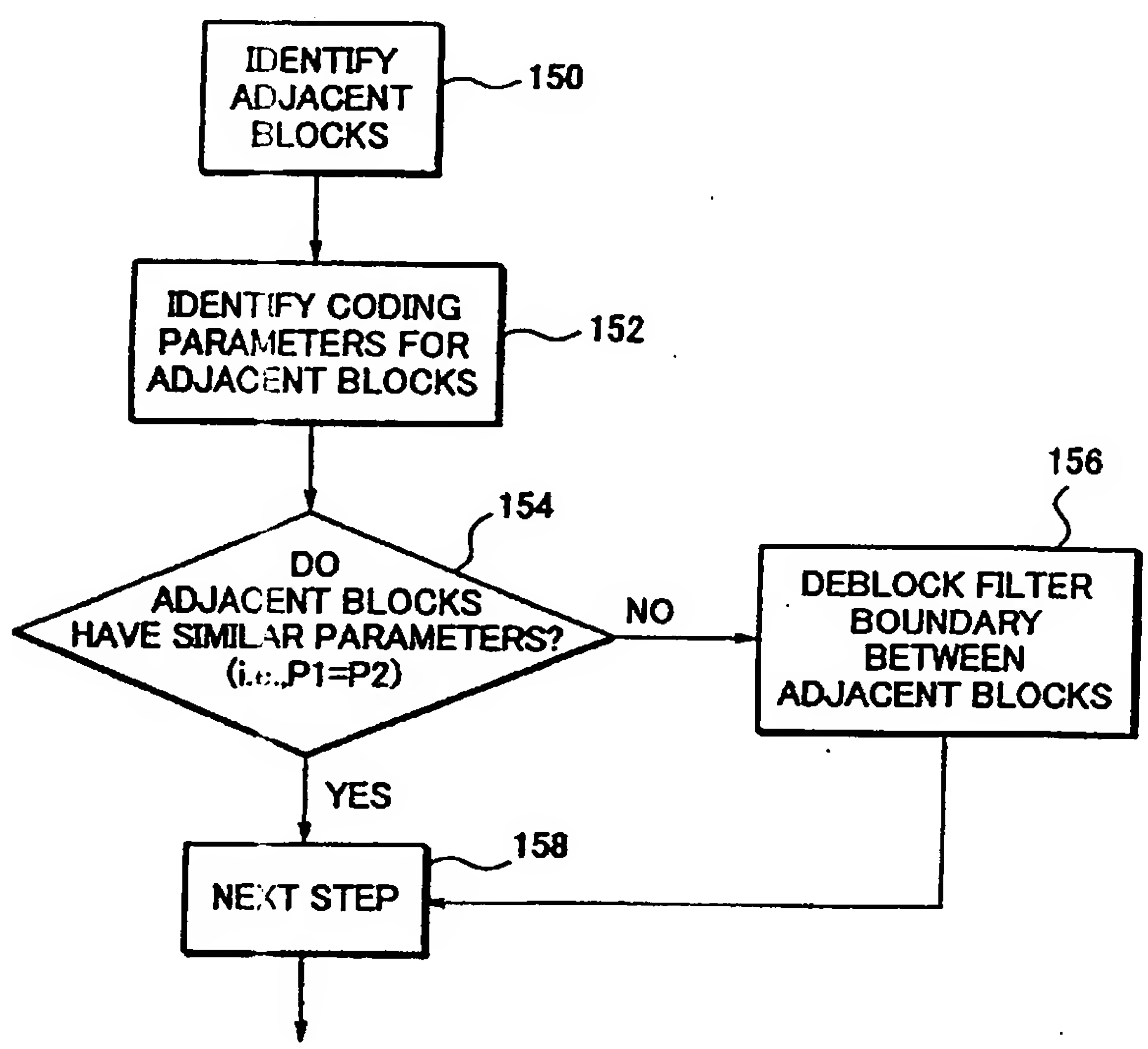
(72) Inventors:
- **Sun, Shijun**
  Vancouver, WA 98683 (US)
- **Lei, Shawmin**
  Camas, WA 98607 (US)

(74) Representative: **Müller - Hoffmann & Partner Patentanwälte,**
Innere Wiener Strasse 17
81667 München (DE)

(54) **Method and apparatus for the reduction of artifact in decompressed images using post-filtering**

(57) Adjacent blocks (44, 46) are identified in an image. Coding parameters for the adjacent blocks (44, 46) are identified (102). Deblock filtering (104) between the identified adjacent blocks (44, 46) is skipped if the coding parameters (55) for the identified adjacent blocks (44, 46) are similar and not skipped if the coding parameters (55) for the identified adjacent blocks (44, 46) are substantially different.

FIG.8



EP 1 246 131 A2

Printed by Jouve, 75001 PARIS (FR)

Description

BACKGROUND

[0001] Block based motion compensated video coding is used in many video compression standards such as H.261, H.263, H263+, MPEG-1, MPEG-2, and H26L. The lossy compression process can create visual artifacts in the decoded images, referred to as image artifacts. Blocking artifacts occur along the block boundaries in an image and are caused by the coarse quantization of transform coefficients.

[0002] Image filtering techniques can be used to reduce artifacts in reconstructed images. Reconstructed images are the images produced after being inverse transformed and decoded. The rule of thumb in these techniques is that image edges should be preserved while the rest of the image is smoothed. Low pass filters are carefully chosen based on the characteristic of a particular pixel or set of pixels surrounding the image edges.

[0003] Non-correlated image pixels that extend across image block boundaries are specifically filtered to reduce blocking artifacts. However, this filtering can introduce blurring artifacts into the image. If there are little or no blocking artifacts between adjacent blocks, then low pass filtering needlessly incorporates blurring into the image while at the same time wasting processing resources.

[0004] The present invention addresses this and other problems associated with the prior art.

SUMMARY OF THE INVENTION

[0005] Adjacent blocks are identified in an image. Coding parameters for the adjacent blocks are identified. Deblock filtering between the identified adjacent blocks is skipped if the coding parameters for the identified adjacent blocks are similar and not skipped if the coding parameters for the identified adjacent blocks are substantially different.

BRIEF DESCRIPTION OF THE DRAWINGS

[0006] FIG. 1 is a diagram showing how deblock filtering is selectively skipped according to similarities between adjacent image blocks.

[0007] FIG. 2 is a diagram showing two adjacent image blocks having similar motion vectors.

[0008] FIG. 3 is a diagram showing how transform coefficients are identified for one of the image blocks.

[0009] FIG. 4 is a diagram showing how residual transform coefficients are compared between two adjacent image blocks.

[0010] FIG. 5 is a block diagram showing how the video image is encoded and decoded.

[0011] FIG. 6 is a block diagram showing how deblock filtering is selectively skipped in a codec.

[0012] FIG. 7 shows a table containing the results from selective deblock filter skipping.

[0013] FIG. 8 is a flow chart describing the steps of an embodiment of the present invention in which deblock filtering between adjacent blocks is dependent on similarity of coding parameters in adjacent blocks.

[0014] FIG. 9 is a flow chart describing the steps of an embodiment of the present invention in which deblock filtering between adjacent blocks is dependent on adjacent blocks having similar motion vectors.

[0015] FIG. 10 is a flow chart describing the steps of an embodiment of the present invention in which deblock filtering between adjacent blocks is dependent on adjacent blocks having similar motion vectors that point to the same reference frame.

[0016] FIG. 11 is a flow chart describing the steps of an embodiment of the present invention in which deblock filtering between adjacent blocks is dependent on adjacent blocks having similar motion vectors that point to adjacent reference blocks in a single reference frame.

[0017] FIG. 12 is a flow chart describing the steps of an embodiment of the present invention in which deblock filtering between adjacent blocks is dependent on adjacent blocks having parameters comprising similar D.C. transform coefficients.

[0018] FIG. 13 is a flow chart describing the steps of an embodiment of the present invention in which deblock filtering between adjacent blocks is dependent on adjacent blocks having parameters comprising similar A.C. transform coefficients.

[0019] FIG. 14 is a flow chart describing the steps of an embodiment of the present invention in which deblock filtering between adjacent blocks is dependent on adjacent blocks in a luminance image having parameters comprising similar motion vectors and similar motion vector targets in a reference frame.

[0020] FIG. 15 is a flow chart describing the steps of an embodiment of the present invention in which deblock filtering between adjacent blocks is dependent on adjacent blocks in a luminance image having parameters comprising similar motion vectors, similar motion vector targets in a reference frame and similar transform coefficients.

[0021] FIG. 16 is a flow chart describing the steps of an embodiment of the present invention in which an image is split into separate luminance and chrominance channels and deblock filtering between adjacent blocks in each luminance or chrominance image is dependent on adjacent blocks in a luminance image having parameters comprising similar motion vectors.

[0022] FIG. 17 is a flow chart describing the steps of an embodiment of the present invention in which an image is split into separate luminance and chrominance channels and deblock filtering between adjacent blocks in each luminance or chrominance image is dependent on adjacent blocks in a luminance image having parameters comprising similar motion vectors, similar motion vector targets in a reference frame and similar transform

coefficients.

DETAILED DESCRIPTION

[0023] In conventional filtering methods, filter processing only considers a single reconstructed image frame at a time. The motion-vector information available at both the encoder and decoder is not used. If two adjacent blocks share the same motion vector with respect to the same reference image frame, (for a multiple reference frames system) there may be no significant difference between the image residuals of each block. The block boundary of these two adjacent blocks may have been filtered in the reference frame and should therefore not be filtered again for the current frame. If a deblock filter is used without considering this motion-vector information, the conventional filtering process might filter the same boundary again and again from frame to frame. This unnecessary filtering not only causes unnecessary blurring but also results in extra filter computations.

[0024] FIG. 1 shows an image 12 that selectively filters block artifacts according to similarities between image blocks. The image 12 includes multiple image blocks 14 that are each individually encoded before being stored, transmitted, etc. The borders between some of the blocks 14 include blocking artifacts 18. Blocking artifacts are any image discontinuities between blocks 14 that may be created by the encoding process. A low pass filter is used to reduce the blocking artifacts that exist at the borders of adjacent image blocks.

[0025] For example, blocking artifacts 24 exist between blocks 20 and 22. A low pass filter is used at the border 26 between blocks 20 and 22 to remove or reduce the blocking artifacts 24. The low pass filter in one example selects a group of pixels 28 from both sides of the border 26. An average pixel value is derived from the group of pixels 28. Then each individual pixel is compared to the average pixel value. Any pixels in group 28 outside of a predetermined range of the average pixel value is then replaced with the average pixel value.

[0026] As described above, if there are little or no blocking artifacts 24 between the adjacent pixels, then the group of pixels 28 may be needlessly filtered causing blurring in the image and wasting processing resources. A skip mode filtering scheme uses the motion estimation and compensation information for adjacent image blocks. If the motion estimation and compensation information is similar, deblock filtering is skipped. This not only avoids unnecessary image blurring but also significantly reduces the required number of filtering operations.

[0027] For example, it is determined during the encoding process that adjacent image blocks 30 and 32 have similar coding parameters. Accordingly, deblock filtering is skipped for the groups of pixels 34 that extend across the border 31 between adjacent blocks 30 and 32. Skip mode filtering can be used for any horizontal or vertical boundary between any adjacent blocks in image 12.

[0028] FIG. 2 shows reference frames 42 and 48 and a current frame 40 that is currently being encoded or decoded. Coding parameters for blocks 44 and 46 are compared to determine whether deblock filtering should be skipped between the two adjacent blocks 44 and 46. One encoding parameter that is compared is the Motion Vectors (MV) for the blocks 44 and 46.

[0029] The motion vector MV1 points from block 44 in current image frame 40 to an associated block 44' in the reference frame 42. The motion vector MV2 points from block 46 in current image frame 40 to an associated block 46' in reference frame 42. Skip mode filtering checks to see if the motion vectors MV1 and MV2 point to adjacent blocks in the same reference frame 42. If the motion vectors point to adjacent blocks in the reference frame (MV1=MV2), then deblock filtering is skipped. This motion vector information may be used along with other coding information to decide whether to skip deblock filtering between the two image blocks 44 and 46.

[0030] More than one reference frame may be used during the encoding and decoding process. For example, there may be another reference frame 48. The adjacent blocks 44 and 46 may have motion vectors pointing to different reference frames. In one embodiment, the decision to skip deblock filtering depends on whether the motion vectors for the two adjacent blocks point to the same reference frame. For example, image block 44 may have a motion vector 49 pointing to reference frame 48 and image block 46 may have the motion vector MV2 pointing to reference frame 42. Deblock filtering is not skipped in this example because the motion vectors 49 and MV2 point to different reference frames.

[0031] FIG. 3 shows another coding parameter that may be used to decide whether or not to skip deblock filtering. The image block 44 from image frame 40 is compared with reference block 44' from the reference frame 42 pointed to by the motion vector MV1 as previously shown in FIG. 2. A residual block 44" is output from the comparison between image block 44 and reference block 44'. A transform 50 is performed on the residual block 44" creating a transformed block 44"' of transform coefficients. In one example, the transform 50 is a Discrete Cosine Transform. The transformed block 44"' includes a D.C. component 52 and A.C. components 53.

[0032] The D.C. component 52 refers to a lowest frequency transform coefficient in image block 44. For example, the coefficient that represents the average energy in the image block 44. The A.C. components 53 refer to the transform coefficients that represent the higher frequency components in the image block 44. For example, the transform coefficients that represent the large energy differences between pixels in the image block 44.

[0033] FIG. 4 shows the transformed residual blocks 44"' and 46"'. The D.C. components 52 from the two

transformed blocks 44"' and 46"' are compared in processor 54. If the D.C. components are the same or within some range of each other, the processor 54 notifies a deblock filter operation 56 to skip deblock filtering between the border of the two adjacent blocks 44 and 46. If the D.C. components 52 are not similar, then no skip notification is initiated and the border between blocks 44 and 46 is deblock filtered.

[0034] In one embodiment, skip mode filtering is incorporated into the Telecommunications Sector of the International Telecommunication Union (ITU-T) proposed H.26L encoding scheme. The H.26L scheme only uses 4x4 integer Discrete Cosine Transform (DCT) blocks. Here, only the D.C. component of the two adjacent blocks may be checked. However some limited low frequency A.C. coefficients could also be checked when the image blocks are bigger sizes, such as 8x8 or 16x16 blocks. For example, the upper D.C. component 52 and the three lower frequency A.C. transform coefficients 53 for block 44"' may be compared with the upper D.C. component 52 and three lower frequency A.C. transform coefficients 53 for block 46"'. Different combinations of D.C. and/or low frequency A.C. transform coefficients can be used to identify the relative similarity between the two adjacent blocks 44 and 46.

[0035] The processor 54 can also receive other coding parameters 55 that are generated during the coding process. These coding parameters include the motion vectors and reference frame information for the adjacent blocks 44 and 46 as described above. The processor 54 uses all of these coding parameters to determine whether or not to skip deblock filtering between adjacent image blocks 44 and 46. Other encoding and transform functions performed on the image may be carried out in the same processor 54 or in a different processing circuit. In the case where all or most of the coding is done in the same processor, the skip mode is simply enabled by setting a skip parameter in the filtering routine.

[0036] FIG. 5 shows how skip mode filtering is used in a block-based motion-compensated Coder-Decoder (Codec) 60. The codec 60 is used for inter-frame coding. An input video block from the current frame is fed from box 62 into a comparator 64. The output of a frame buffering box 80 generates a reference block 81 according to the estimated motion vector (and possible reference frame number). The difference between the input video block and the reference block 81 is transformed in box 66 and then quantized in box 68. The quantized transform block is encoded by a Variable Length Coder (VLC) in box 70 and then transmitted, stored, etc.

[0037] The encoding section of the codec 60 reconstructs the transformed and quantized image by first Inverse Quantizing (IQ) the transformed image in box 72. The inverse quantized image is then inverse transformed in box 74 to generate a reconstructed residual image. This reconstructed residual block is then added in box 76 to the reference block 81 to generate a reconstructed image block. Generally the reconstructed image is loop filtered in box 78 to reduce blocking artifacts caused by the quantization and transform process. The filtered image is then buffered in box 80 to form reference frames. The frame buffering in box 80 uses the reconstructed reference frames for motion estimation and compensation. The reference block 81 is compared to the input video block in comparator 64. An encoded image is output at node 71 from the encoding section and is then either stored or transmitted.

[0038] In a decoder portion of the codec 60, a variable length decoder (VLD) decodes the encoded image in box 82. The decoded image is inverse quantized in box 84 and inverse transformed in box 86. The reconstructed residual image from box 86 is added in the summing box 88 to the reference block 91 before being loop filtered in box 90 to reduce blocking artifacts and buffered in box 92 as reference frames. The reference block 91 is generated from box 92 according to the received motion vector information. The loop filtered output from box 90 can optionally be post filtered in box 94 to further reduce image artifacts before being displayed as a video image in box 96. The skip mode filtering scheme can be performed in any combination of the filtering functions in boxes 78, 90 and 94.

[0039] The motion estimation and compensation information available during video coding are used to determine when to skip deblock filtering in boxes 78, 90 and/or 94. Since these coding parameters are already generated during the encoding and decoding process, there are no additional coding parameters that have to be generated or transmitted specially for skip mode filtering.

[0040] FIG. 6 shows is further detail how skip mode filtering is used in the filters 78, 90, and/or 94 in the encoder and decoder in FIG. 5. The interblock boundary between any two adjacent blocks "j" and "k" is first identified in box 100. The two blocks may be horizontally or vertically adjacent in the image frame. Decision box 102 compares the motion vector mv(j) for block j with the motion vector mv(k) for block k. It is first determined whether the two adjacent blocks j and k have the same motion vector pointing to the same reference frame. In other words, the motion vectors for the adjacent blocks point to adjacent blocks (mv(j) = mv(k)) in the same reference frame (ref(j) = ref(k)).

[0041] It is then determined whether the residual coefficients for the two adjacent blocks are similar. If there is no significant difference between the image residuals of the adjacent blocks, for example, the two blocks j and k have the same or similar D.C. component (dc(j) = dc(k)), then the deblock filtering process in box 104 is skipped. Skip mode filtering then moves to the next interblock boundary in box 106 and conducts the next comparison in decision box 102. Skip mode filtering can be performed for both horizontally adjacent blocks and vertically adjacent blocks.

[0042] In one embodiment, only the reference frame and motion vector information for the adjacent image

blocks are used to determine block skipping. In another embodiment only the D.C. and/or A.C. residual coefficients are used to determine block skipping. In another embodiment the motion vector, reference frame and residual coefficients are all used to determine block skipping.

[0043] The skip mode filtering scheme can be applied to spatially sub-sampled chrominance channels. For example in a case with 4:2:0 color format sequences, skip mode filtering for block boundaries may only rely on the equality of motion vectors and D.C. components for the luminance component of the image. If the motion vectors and the D.C. components are the same, deblock filtering is skipped for both the luminance and chrominance components of the adjacent image blocks. In another embodiment, the motion vectors and the D.C. components are considered separately for each luminance and chrominance component of the adjacent blocks. In this case, a luminance or chrominance component for adjacent blocks may be deblock filtered while the other luminance or chrominance components for the same adjacent blocks are not deblock filtered.

[0044] FIG. 7 includes a table 110 showing the results of skip mode filtering using a ITU-TH.26L Testing Model-Long TML5. Table 110 compares the results of the TML filtering standard with skip mode filtering as described above. Encoding results using skip mode filtering are shown in table 110 under the heading SLA.

[0045] There were four images that were tested, Akiyo_cif for 300 frames at 30 Frames Per Second (fps), Foreman_cif for 300 frames at 30 fps, Foreman_qcif for 100 frames at 10 fps, and Tempete_cif for 260 frames at 30 fps. Quantization Parameters (QP) of 25 and 30 were used. The results show no significant visual quality degradation with the skip mode filtering. The Picture Signal to Noise Ratio (PSNR) for the images stays approximately the same for the luminance Y and chrominance U and V channels. However, skip mode filtering provides time savings of 40-70 percent.

[0046] Skip mode filtering can be used with any system that encodes or decodes multiple image frames. For example, DVD players, video recorders, or any system that transmits image data over a communications channel, such as over television channels or over the Internet.

[0047] The skip mode filtering described above can be implemented with dedicated processor systems, micro controllers, programmable logic devices, or microprocessors that perform some or all of the operations. Some of the operations described above may be implemented in software and other operations may be implemented in hardware.

[0048] For the sake of convenience, the operations are described as various interconnected functional blocks or distinct software modules. This is not necessary, however, and there may be cases where these functional blocks or modules are equivalently aggregated into a single logic device, program or operation with unclear boundaries. In any event, the functional blocks and software modules or described features can be implemented by themselves, or in combination with other operations in either hardware or software.

[0049] Some embodiments of the present invention may be described with reference to Figure 8. In these systems and methods, adjacent blocks 150 in a video frame are identified and coding parameters for these adjacent blocks are identified. The coding parameters for the adjacent blocks are then compared to determine their similarity 154. When the coding parameters are not similar, a deblock filter 156 is applied along the boundary between the adjacent blocks. When the coding parameters are similar, deblock filtering is skipped and the process proceeds to the next step 158. Likewise, when deblock filtering is performed, the process proceeds to the next step 158 after filtering.

[0050] In some embodiments of the present invention, as shown in Figure 9, the coding parameters are motion vectors. In these embodiments, adjacent blocks in a video frame are identified 160 and coding parameters 162 comprising motion vectors are identified. These motion vectors are compared to determine their similarity 164. When the motion vectors are not similar, deblock filtering may be performed 166 between the adjacent blocks and the process may proceed to its next step 168. When the motion vectors are similar, deblock filtering is skipped and the next step 168 is accomplished directly.

[0051] Other embodiments of the present invention, as shown in Figure 10, may use multiple coding parameters to determine whether to skip filtering. In these embodiments, adjacent blocks are identified 170 and coding parameters 172 are determined for the adjacent blocks. These coding parameters may comprise motion vector attributes including the target frame of the motion vectors. When motion vectors of adjacent blocks are not similar 174, deblock filtering may be performed 176 between the adjacent blocks. When motion vectors are similar 174, other parameters may be used to further qualify the filtering process. In this example, the motion vectors may be compared to determine whether they point to the same reference frame 178. If the vectors do not point to the same reference frame, deblock filtering may be performed between the blocks 176. If the vectors point to the same reference frame, filtering may be skipped and the process may proceed to the next step 179.

[0052] Further motion vector parameters may be used to determine filtering. In embodiments illustrated in Figure 11, the location of the blocks to which vectors point is a parameter that may be used to determine filtering options. In these embodiments, adjacent blocks are identified 200 and coding parameters are identified for the adjacent blocks 202. Motion vectors are then compared to determine their similarity 204. If the vectors are not similar, deblock filtering may proceed 208. If motion vectors are similar, another comparison may be made to determine whether the motion vectors of the

adjacent blocks point to the same reference frame. If the vectors don't point to the same frame, deblock filtering may proceed 208. If the vectors do point to the same reference frame, the blocks to which the vectors point may be compared 210. When motion vectors do not point to adjacent blocks in the same reference frame, deblock filtering may proceed 208. When the vectors point to adjacent blocks in the same reference frame, deblock filtering may be skipped and a next step 212 may be executed. In this manner, adjacent blocks which reference adjacent blocks in a reference frame and which are not likely to have significant artifacts therebetween are not deblock filtered. This deblock filter skipping avoids any blurring and image degradation caused by the filtering process. Processing time is also conserved as unnecessary filtering is avoided. Image quality is thereby improved and fewer calculations are required in the process. It should be noted that various combinations of these motion vector parameters may be used to determine filter skipping. These myriad combinations are not specifically described in detail, but are thought to be within the grasp of one skilled in the art and are intended to fall within the scope of the appended claims.

[0053] Further embodiments of the present invention may utilize transform coefficients to determine whether deblock filtering should occur. In reference to Figure 12, adjacent blocks 180 in a frame are identified and coding parameters are identified for the adjacent blocks 182. These coding parameters may comprise motion vector parameters as well as transform coefficients.

[0054] Motion vectors are then compared 184 to determine similarity. If the motion vectors are not similar, deblock filtering may be performed 186. If the motion vectors are similar, the motion vector data is analyzed to determine whether the motion vectors point to the same reference frame. If the motion vectors do not point to the same reference frame 185, filtering may proceed 186.

[0055] If the motion vectors point to the same reference frame 185, transform coefficients may be compared to further qualify filtering processes. In this example, DC transform coefficients obtained through Discrete Cosine Transform (DCT) methods or other methods may be compared for the adjacent blocks. If the DC transform coefficients are not similar 187, deblock filtering may be performed 186. If the DC transform coefficients are similar, filtering may be skipped and the methods and systems may proceed to the next step 188.

[0056] Still other embodiments of the present invention may utilize AC transform coefficients to determine filtering options. In reference to Figure 13, embodiments similar to those described in relation to Figure 12 are illustrated with the additional steps of evaluating AC transform coefficients. In these embodiments, blocks 190 and their coding parameters 191 are identified. Similarities in motion vectors 192, motion vector target frames 193 and DC transform coefficients are also compared 194. When similarities in these parameters exist, AC transform coefficients are compared 196 and, if they are similar, deblock filtering is skipped and the next step in the process is executed 197. If the AC coefficients are not similar, filtering is performed between the adjacent blocks and the process proceeds on to the next step 197.

[0057] AC transform coefficients are more likely to have significance in larger blocks, but can be used in methods utilizing smaller blocks such as 4x4 blocks.

[0058] In some embodiments of the present invention, an image may be separated into various luminance and chrominance channels depending on the format of the image and the color space utilized. In the following examples, a YUV color space is described, however, many other formats and color spaces may be used in these embodiments. CieLAB, YcrCb and other spaces may be used. In alternative embodiments color spaces such as RGB may be used.

[0059] Some embodiments of the present invention may be described in relation to Figure 14. In these embodiments, luminance data is extracted from the image and a luminance image is created 220. Adjacent blocks are then identified in the luminance image 222 and coding parameters for the adjacent blocks are also identified 224. As in other embodiments, the motion vectors of the adjacent blocks are compared to determine similarities 226. When the motion vectors are not similar, deblock filtering is performed 230, when the vectors are similar further analysis is performed to determine whether the vectors point to the same reference frame 228. When the vectors point to different reference frames, deblock filtering is performed between the adjacent blocks 230 of the original image that correspond to the adjacent blocks in the luminance image. When the vectors point to the same frame, deblock filtering is skipped and the next step is executed without prior filtering 232. When filtering is performed, the next step is executed 232 after the filtering processes. Accordingly, analysis of data in the luminance channel is used to determine filtering processes in the original image, which contains both luminance and chrominance data.

[0060] In other related embodiments, illustrated in Figure 15, a luminance image is created 240 and corresponding adjacent blocks are identified in the luminance and original image 242. Coding parameters are also identified for the luminance image blocks 244. Subsequently, motion vectors are compared to determine similarities 246. If significant similarities do not exist, filtering is performed between the adjacent blocks in the original image 252. If motion vectors are similar, the target frames of the motion vectors are compared to determine whether the vectors point to the same reference frame. If the vectors do not point to the same reference frame, filtering is performed. If the vectors point to the same reference frame, transform coefficients of the luminance (Y) image are compared. If transform coefficients are not similar, filtering is performed. If transform coeffi-

cieni ıre similar, filtering is skipped and the next step 254 i ·xecuted. Likewise, the next step is executed 254 after ·. ıy filtering operation.

[006 Images may be further divided into component char· ·ls that generally correspond to luminance and chro· ıance channels. In some embodiments of the pres : invention, each channel may be filtered according t· ·arameters unique to that channel.

[006· As an example, embodiments may be describ ·' with reference to Figure 16, wherein an image is di ·led into separate luminance (Y) and multiple chro ıance (U, V) channels 260. In these embodimen. adjacent blocks are identified in images correspor.. ıg to each channel 262, 272, 282. Coding parameters :uch as motion vectors data, are also identified for tl ;e blocks in each channel 264, 274, 284. These codir parameters may then be compared to determine simil· ·ties as in other embodiments. In these exemplary emb ··iments, motion vector similarities for channel-spec : motion vectors may be used to determine filtering c. ·ions in each channel. When the motion vectors for a : :annel image are not similar 266, 276, 286, filtering i. ·erformed in that specific channel between the adja(.ıt blocks 270, 280, 290. If the motion vectors are simil. · the target reference frames are compared 268, 278, 38. When the vectors for adjacent blocks in a chan· ·l point to the same reference frame, filtering is skip| .:. When the vectors point to different reference fram filtering is performed 270, 280, 290.

[006 As in other embodiments, these channelized emb ·:ments may utilize transform coefficient data to quali , filtering options. As shown in Figure 17, the methods ; d systems described in relation to Figure 16 may furth compare channel transform coefficients 310, 322, :4. When the coefficients are not similar, filtering is pe ·rmed 312, 324, 336. When the coefficients are simil filtering is skipped.

[006 It should be noted that various combinations of para .:ters may be employed in qualifying filtering operati ·s in each channel. DC and AC transform coefficient ·nay be utilized for these embodiments. Furthermor(/arious channels and combinations of channels may · · used to determine filtering options and perform filter· :. For example, both chrominance channels may be c ıbined and analyzed together in some embodimen . Data and parameters from one channel may also be u · d to determine filtering options in another channel. ʳ example, parameters taken from the U chrominanc · channel may be compared to determine filtering optic . in the V chrominance channel and vice versa.

[006 Having described and illustrated the principles of tl invention in various exemplary embodiments ther(. , it should be apparent that the invention may be mod ·:d in arrangement and detail without departing from : ıch principles. Claim is made to all modifications and ·iations coming within the spirit and scope of the follo· ıg claims.

Claims

1. A method f ;elective image filtering, said method comprising

   identify ·ı (100) adjacent blocks in the image;
   identify · ı (102) coding parameters for the adjacent ' cks;
   skippir: :eblock filtering between the identified adjace , ılocks when the identified coding parameter· for the identified adjacent blocks are similar; ıd
   debloc ltering (104) between the identified adjace : ılocks when the identified coding parameter for the identified adjacent blocks are not sim· r.

2. A method ε ording to claim 1 wherein said identified coding · ırameters (55) comprise motion vectors (49, M' MV2).

3. A method ε ording to claim 1 wherein said identified coding rameters (55) comprise reference image frames 2, 48).

4. A method ε· ording to claim 1 wherein said identified coding · rameters (55) comprise transform coefficients.

5. A method ε ording to claim 1 wherein said identified coding ' rameters (55) comprise D.C. components (52) i he transform coefficients.

6. A method ε: ording to claim 1 wherein said identified coding· rameters (55) comprise A.C. components (53) i he transform coefficients.

7. A method ε · ording to claim 1 wherein said identified coding : ırameters (55) comprise at least one parameter ·· .en from the group consisting of motion vector), MV1, MV2) data, reference image frame (42, ‹ . data, reference image block (44', 46') data, D.C. · ıponent (52) data in a transform coefficient an › \.C. component (53) data in a transform coeffi(ıt.

8. A method f· ·erforming one of encoding or decoding an ima(comprising:

   identify · :ȷ (100) adjacent blocks in the image;
   identify · ȷ (102) coding parameters for the adjacent ·cks;
   skippir · !eblock filtering between the identified adjace · blocks when the coding parameters for the ·entified adjacent blocks are similar; and
   debloc · iltering (104) between the identified adjace· blocks when the coding parameters

for the identified adjacent blocks are not similar.

9. A method according to claim 8 including skipping deblock filtering (104) when the adjacent blocks (44, 46) have similar motion vectors (49, MV1, MV2) pointing to a same reference image frame (42).

10. A method according to claim 8 including:

    identifying transform coefficients for the adjacent blocks (44, 46); and
    skipping deblock filtering between the adjacent blocks (44, 46) when the transform coefficients are similar.

11. A method according to claim 10 including:

    identifying D.C. components (52) in the transform coefficients; and
    skipping deblock filtering between the adjacent blocks (44, 46) when the D.C.

    components (52) are the same or similar.

12. A method according to claim 11 including:

    identifying A.C. components (53) in the transform coefficients; and
    skipping deblock filtering between the adjacent blocks (44, 46) when the D.C.

    components (52) and A.C. components (53) are the same or similar.

13. A method according to claim 10 including transforming the adjacent blocks (44, 46) using a Discrete Cosine Transform (50) to generate the transform coefficients.

14. A method according to claim 8 including:

    comparing blocks (44, 46) in the image (12) with reference frames (42, 48);
    transforming the result of the comparison between the reference frames (42, 48) and the blocks (44, 46) in the image into transformed blocks (44''', 46''') having transform coefficients;
    comparing the similarities between the transform coefficients; and
    skipping deblock filtering between adjacent blocks (44, 46) in the image (12) according to the results of the comparison between the transform coefficients.

15. A method according to claim 14 wherein the transform coefficients include D.C. transform coefficients.

16. A method according to claim 8 including controlling deblock filtering for a loop filter (78) in an image coder.

17. A method according to claim 8 including controlling deblock filtering in one or both of a loop filter (90) and a post filter (94) in an image decoder.

18. A method according to claim 8 including:

    identifying similarities between coding parameters (55) in a luminance channel of the adjacent blocks; and
    controlling deblock filtering for both the luminance channel and a chrominance channel in the image according to identified similarities in the luminance channel.

19. A method according to claim 8 including selectively skipping deblock filtering in any one of a H.261, H.263, H263+, MPEG-1, MPEG-2, or H26L encoding standard according to coding parameter (55) similarities between adjacent image blocks (44, 46).

20. A codec for at least one of encoding or decoding an image (12), said codec comprising:

    a block identifier for identifying adjacent blocks (44, 46) in the image (12);
    a parameter identifier for identifying coding parameters (55) for said adjacent blocks (44, 46);
    a comparator for comparing said coding parameters (55) for said adjacent blocks (44, 46); and
    a filter selector for selective enablement of at least one filter, said selector operating in response to a comparison of said coding parameters (55),

    wherein said at least one filter is enabled when said coding parameters (55) are not similar and said at least one filter is disabled when said coding parameters (55) are similar.

21. A codec according to claim 20 wherein said identified coding parameters (55) comprise at least one parameter taken from the group consisting of motion vector (49, MV1, MV2) data, reference image frame (42, 48) data, reference image block (44', 46') data, D.C. component (52) data in a transform coefficient and A.C. component (53) data in a transform coefficient.

22. A method for selective image filtering, said method comprising:

    dividing an original image into a plurality of channels;
    identifying adjacent blocks (44, 46) in at least

one of said channels;
identifying coding parameters [illegible]) in said at least one channel;
comparing said coding parame[illegible]s (55);
filtering said original image betw[illegible]en said adjacent blocks (44, 46) when said [illegible]ding parameters (55) are dissimilar and s[illegible]ping filtering said original image between [illegible]id adjacent blocks (44, 46) when said cod[illegible] parameters (55) are similar.

23. A method for selective image filterin[illegible] said method comprising:

dividing an original image int[illegible] plurality of channels, said channels com[illegible]sing a luminance channel and at least on[illegible]hrominance channel wherein each chann[illegible]omprises a channel image;
identifying adjacent blocks (44,[illegible]) in at least one of said channels;
identifying coding parameters [illegible]) in said at least one channels;
comparing said coding parame[illegible]s (55);
filtering said channel images be[illegible]een said adjacent blocks (44, 46) when said [illegible]ding parameters (55) for said channels are [illegible]ssimilar and skipping said filtering when said [illegible]ding parameters (55) are similar.

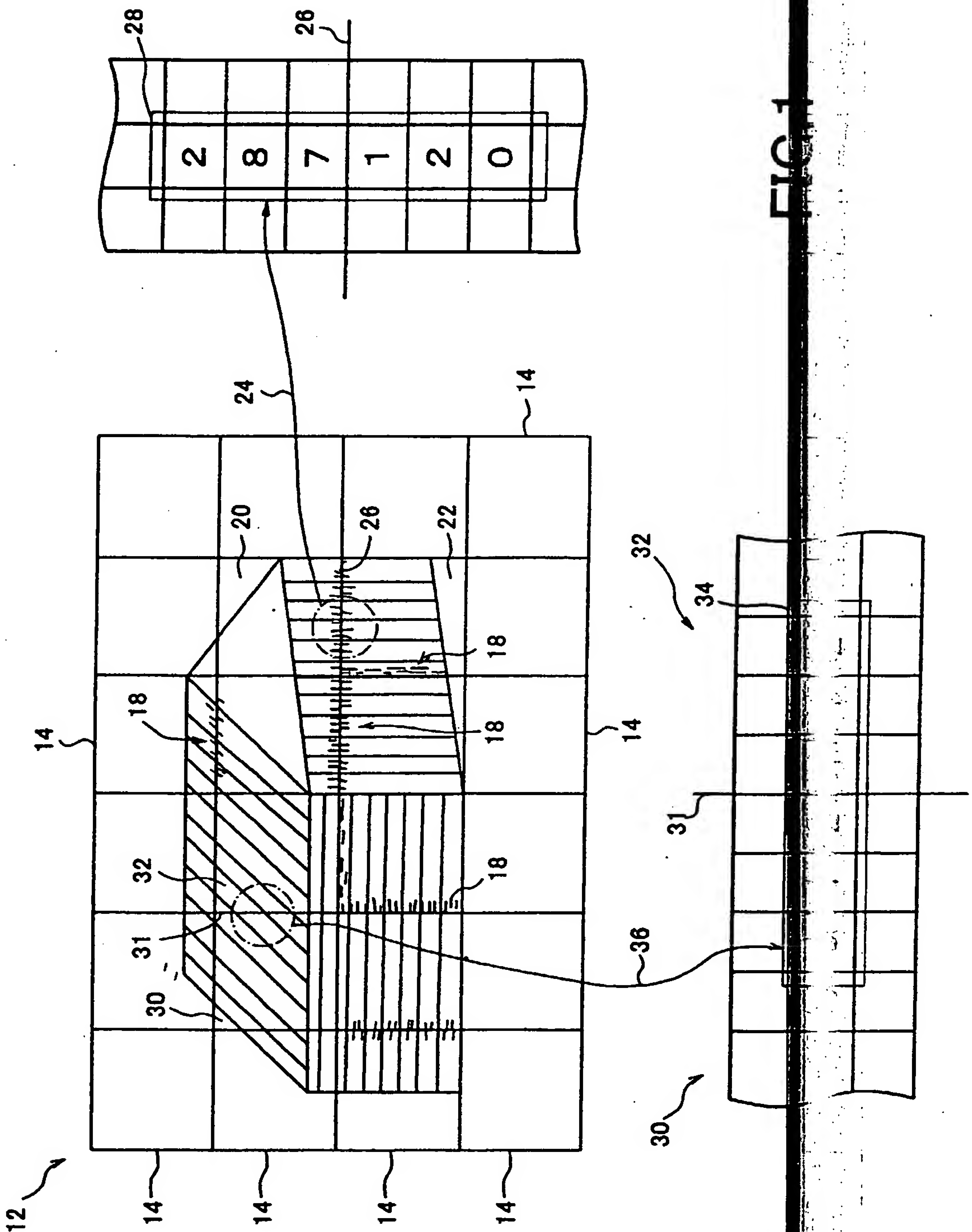

28
26
2
8
7
1
2
0
24
14
20
26
22
18
18
18
18
14
14
32
31
30
12
14
14
14
14
32
34
31
36
30

FIG. 1

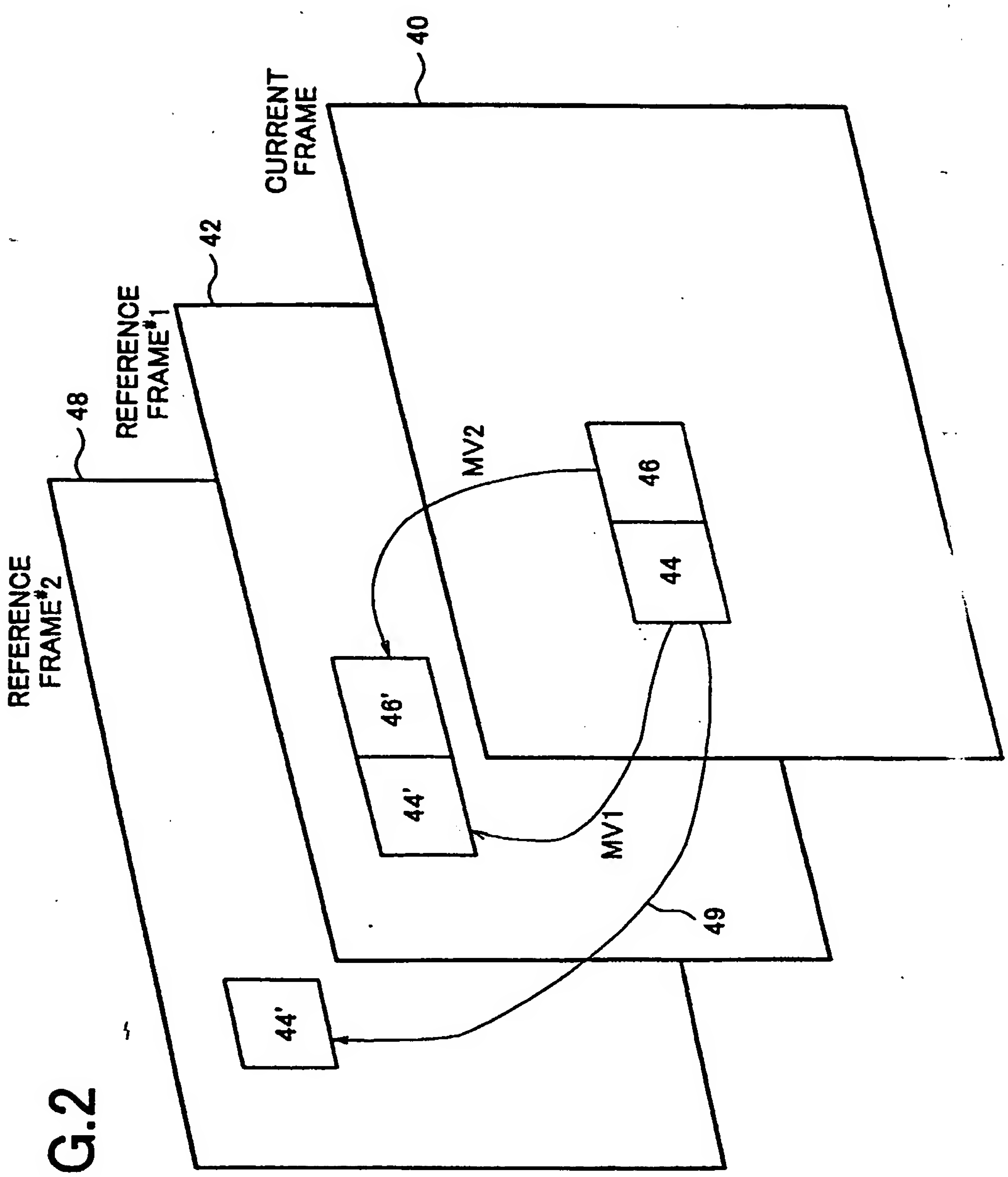
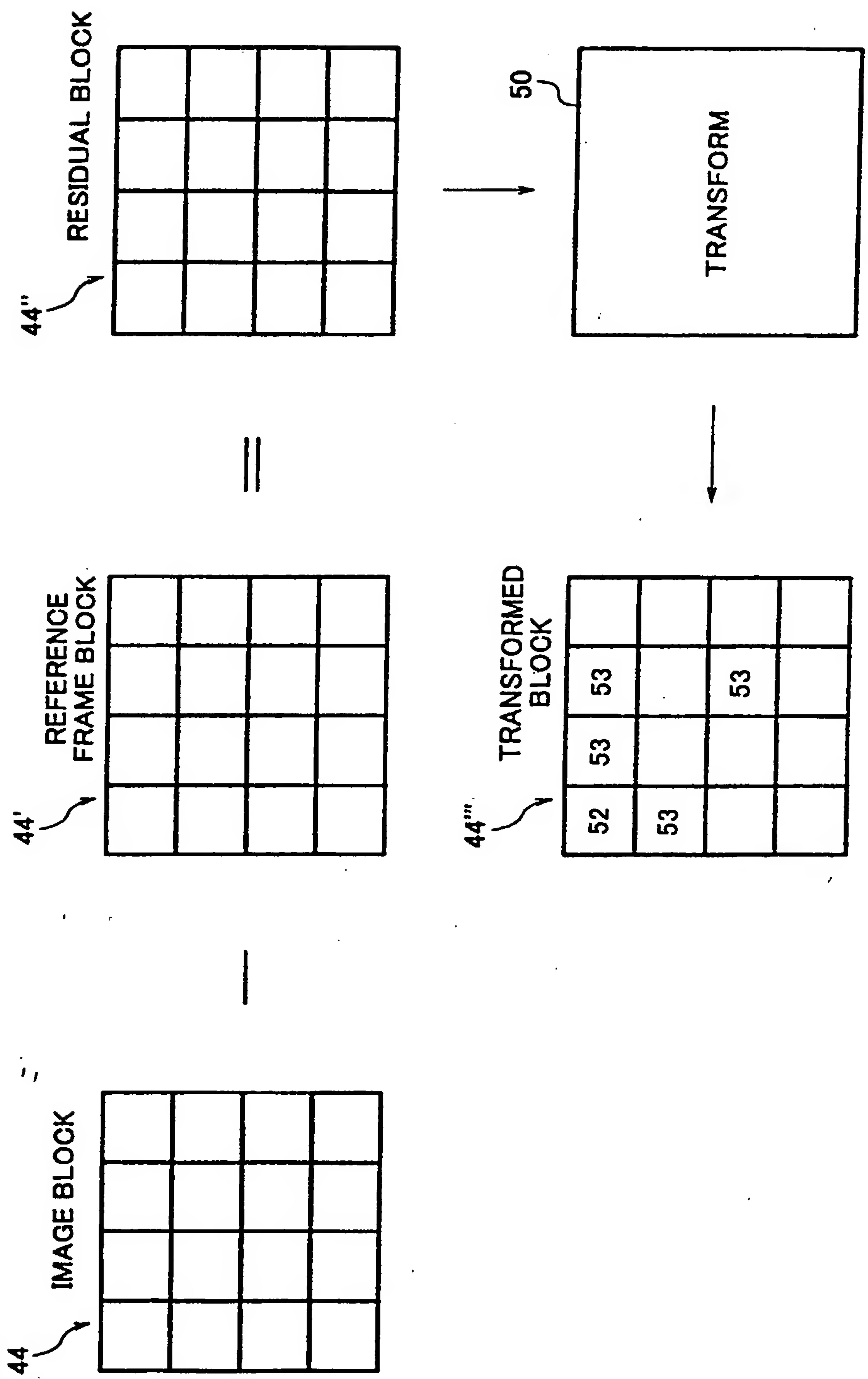

CURRENT FRAME
40
REFERENCE FRAME#1
42
REFERENCE FRAME#2
48
46
44
MV2
46'
44'
MV1
49
44'

FIG.2

FIG.3


44
IMAGE BLOCK
44'
REFERENCE FRAME BLOCK
44''
RESIDUAL BLOCK
50
TRANSFORM
44'''
TRANSFORMED BLOCK
52
53
53
53
53

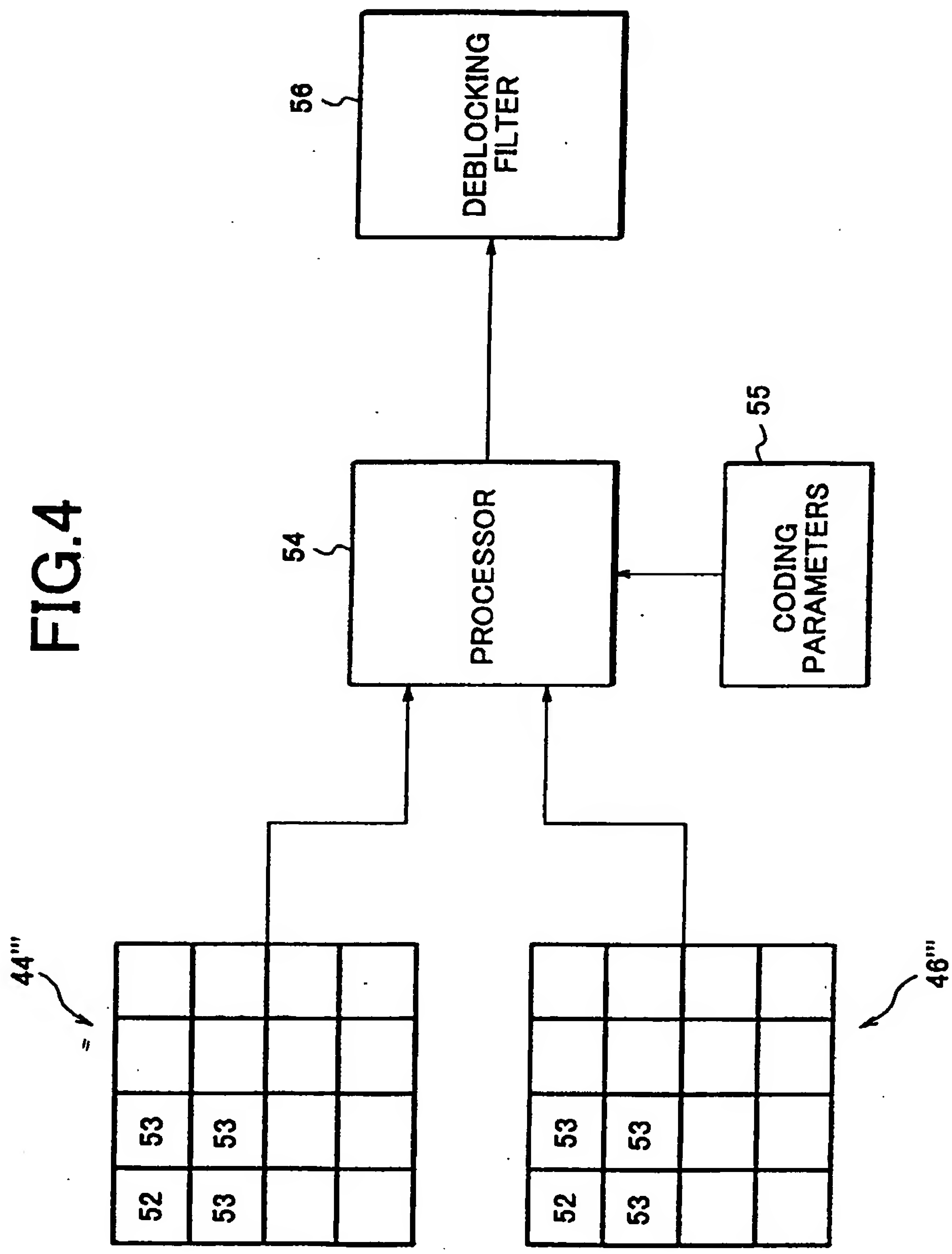

56
DEBLOCKING FILTER
54
PROCESSOR
55
CODING PARAMETERS
44'''
46'''
52
53
53
53
52
53
53
53

FIG.4

FIG.5

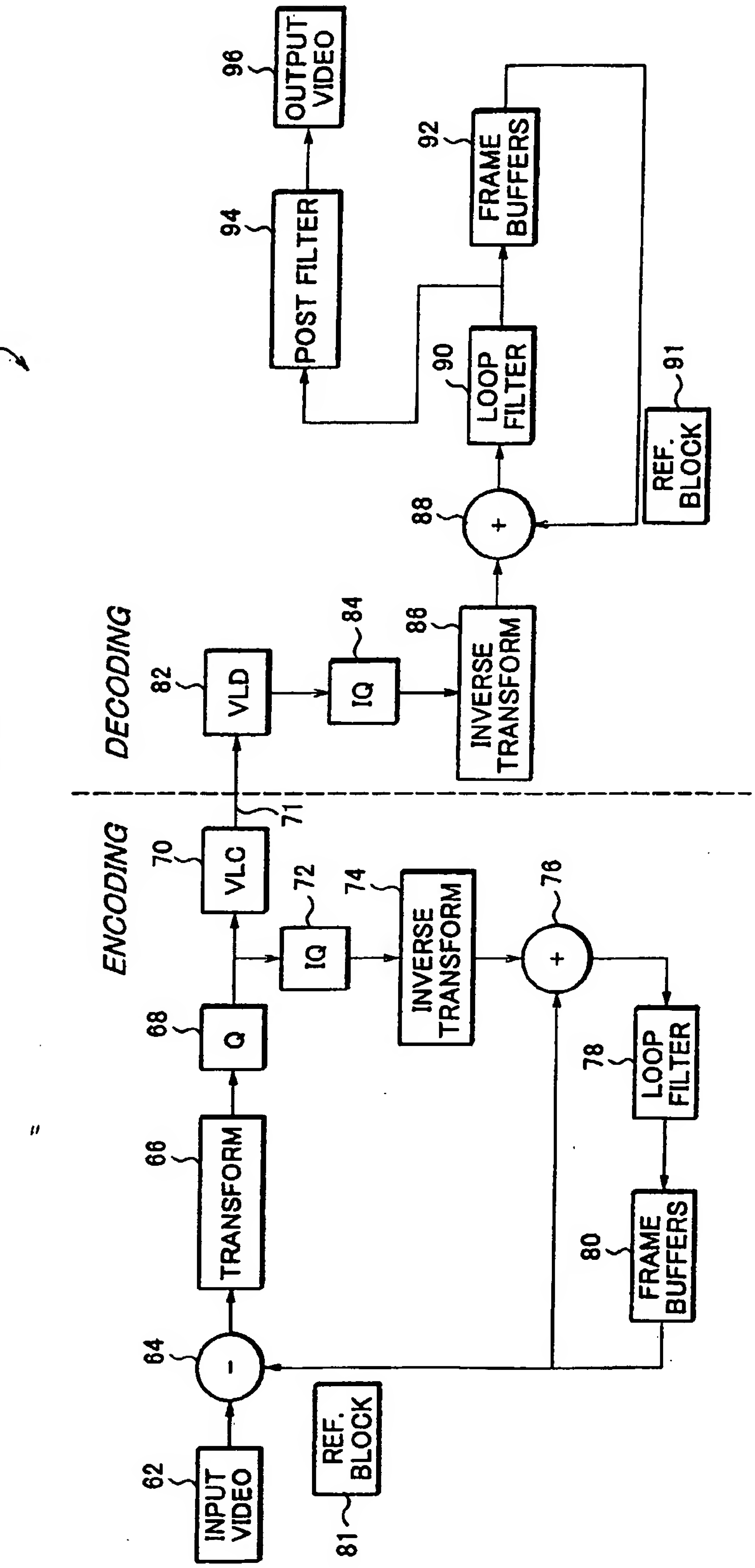

ENCODING
DECODING
INPUT VIDEO
REF. BLOCK
TRANSFORM
Q
IQ
VLC
INVERSE TRANSFORM
LOOP FILTER
FRAME BUFFERS
VLD
IQ
INVERSE TRANSFORM
REF. BLOCK
LOOP FILTER
FRAME BUFFERS
POST FILTER
OUTPUT VIDEO
60
62
64
66
68
70
71
72
74
76
78
80
81
82
84
86
88
90
91
92
94
96

FIG.6

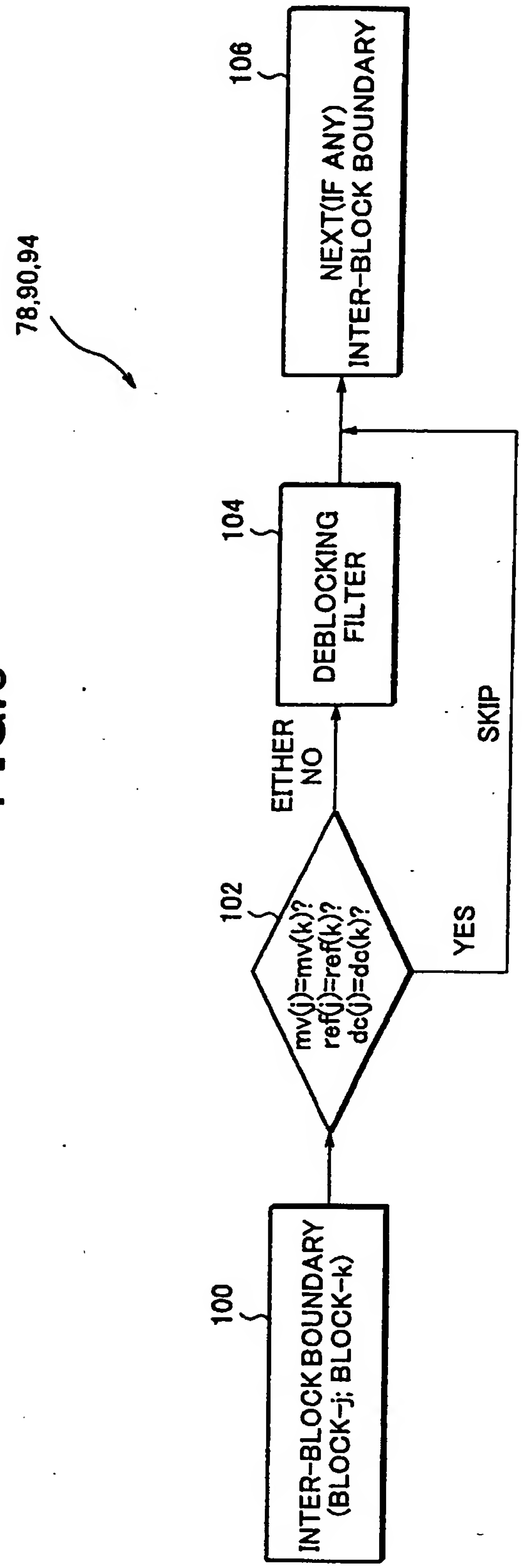

78,90,94
100
INTER-BLOCK BOUNDARY (BLOCK-j; BLOCK-k)
102
mv(j)=mv(k)?
ref(j)=ref(k)?
dc(j)=dc(k)?
EITHER NO
YES
SKIP
104
DEBLOCKING FILTER
106
NEXT(IF ANY) INTER-BLOCK BOUNDARY

FIG.7

110

TABLE 1.COMPARISON BETWEEN TML AND THE PROPOSED LOOP FILTERING SCHEME

VIDEO SEQUENCE	QP	BITRATE(bps)		PSNR(Y)		PSNR(U)		PSNR(V)		NEW LOOP FILTER RELATIVE TITE SAVING
		TML	SLA	TML	SLA	TML	SLA	TML	SLA	
AKIYO_cif 300 FRAMES @ 30fps	25	33151	32346	34.050	34.161	38.934	39.042	40.300	40.369	63.0%
	30	22775	22295	30.797	30.920	36.610	36.964	38.680	38.771	68.2%
FOREMAN_cif 300 FRAMES @ 30fps	25	165115	162740	30.835	31.006	38.124	38.174	38.966	39.030	43.1%
	30	101357	100215	27.580	27.836	36.745	36.811	37.267	37.362	42.1%
FOREMAN_qcif 100 FRAMES @ 10fps	25	28681	28677	29.822	29.931	37.586	37.631	37.773	37.938	38.4%
	30	15999	15822	26.250	26.435	36.432	36.323	36.222	36.283	41.3%
TEMPETE_cif 260 FRAMES @ 30fps	25	336200	329115	28.277	28.490	33.982	34.143	36.009	36.184	45.3%
	30	168133	159789	24.583	24.927	32.334	32.628	34.512	34.799	50.6%

FIG.8

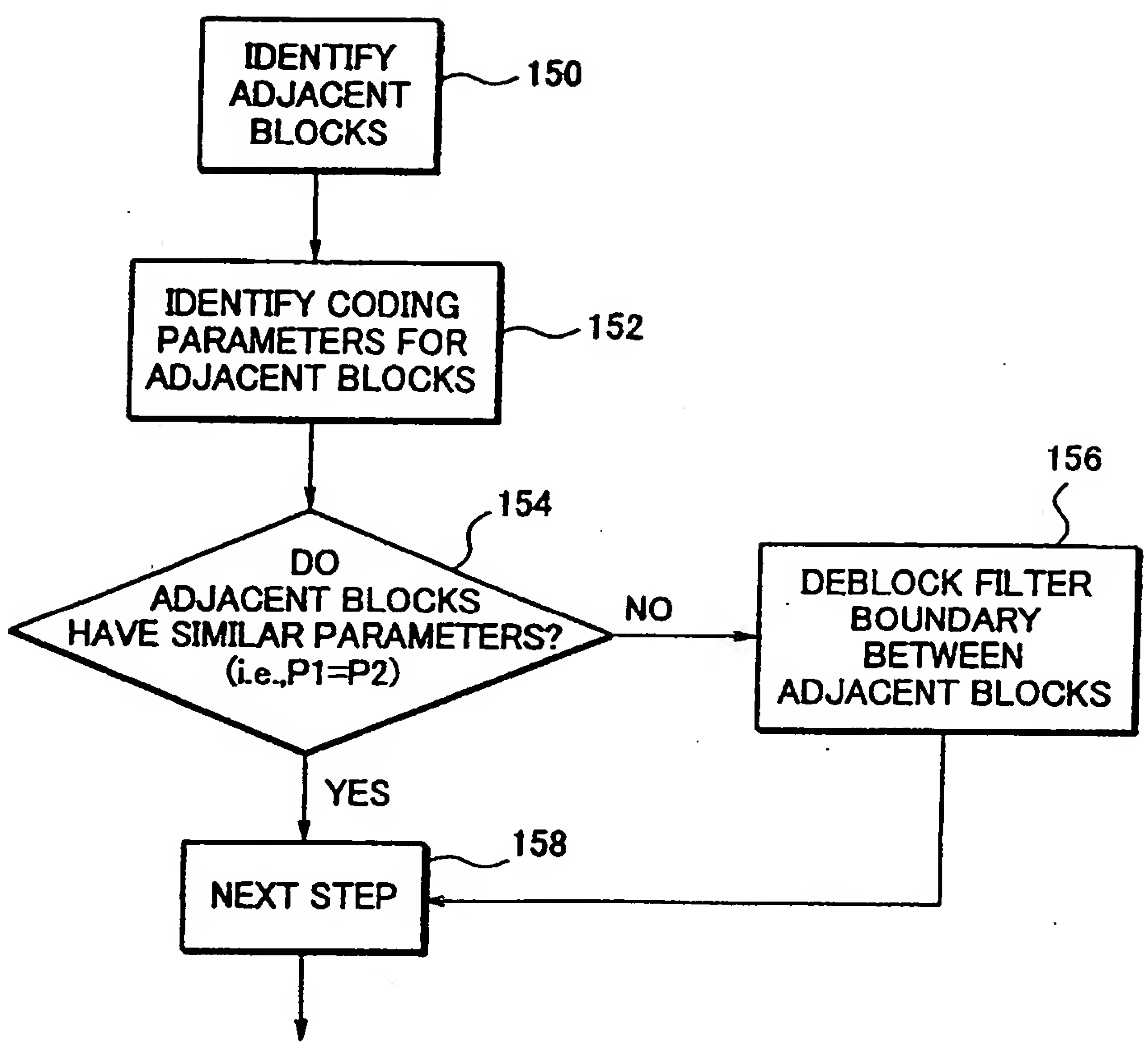

IDENTIFY ADJACENT BLOCKS
150
IDENTIFY CODING PARAMETERS FOR ADJACENT BLOCKS
152
154
DO ADJACENT BLOCKS HAVE SIMILAR PARAMETERS? (i.e.,P1=P2)
NO
156
DEBLOCK FILTER BOUNDARY BETWEEN ADJACENT BLOCKS
YES
158
NEXT STEP

FIG.9

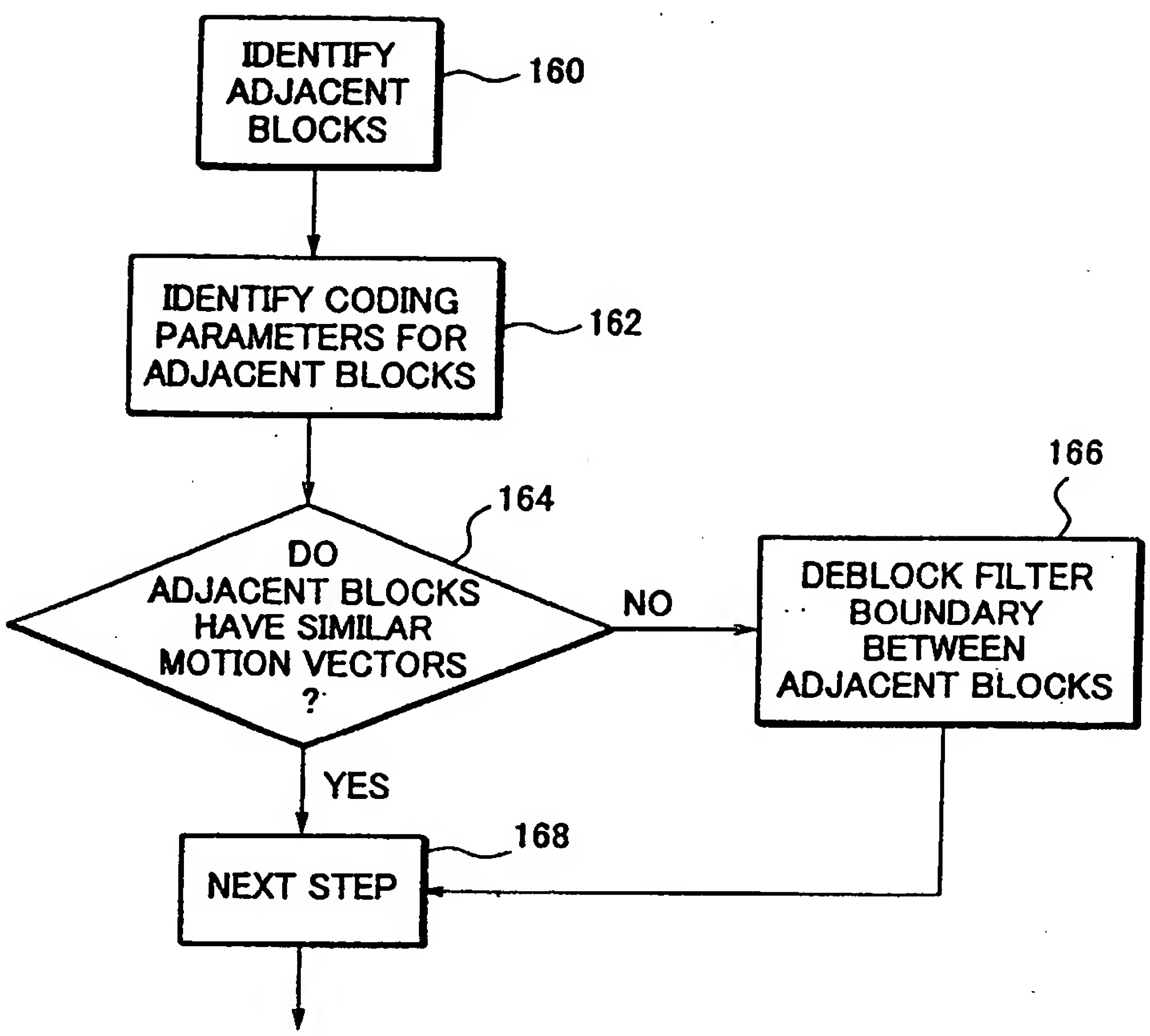

IDENTIFY ADJACENT BLOCKS
160
IDENTIFY CODING PARAMETERS FOR ADJACENT BLOCKS
162
DO ADJACENT BLOCKS HAVE SIMILAR MOTION VECTORS ?
164
NO
DEBLOCK FILTER BOUNDARY BETWEEN ADJACENT BLOCKS
166
YES
NEXT STEP
168

FIG.10

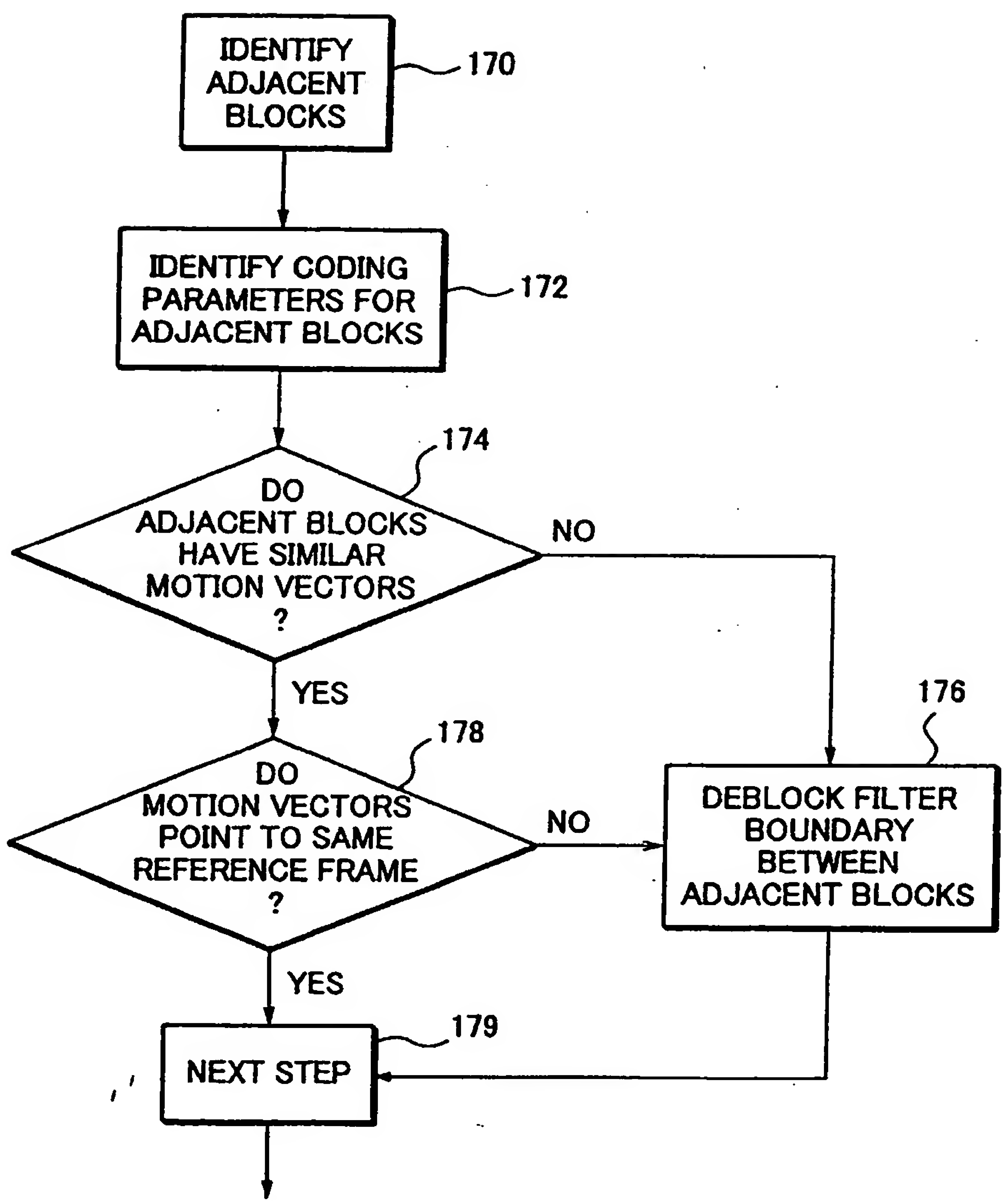

IDENTIFY ADJACENT BLOCKS
170
IDENTIFY CODING PARAMETERS FOR ADJACENT BLOCKS
172
174
DO ADJACENT BLOCKS HAVE SIMILAR MOTION VECTORS ?
NO
YES
176
178
DO MOTION VECTORS POINT TO SAME REFERENCE FRAME ?
NO
DEBLOCK FILTER BOUNDARY BETWEEN ADJACENT BLOCKS
YES
179
NEXT STEP

FIG.11

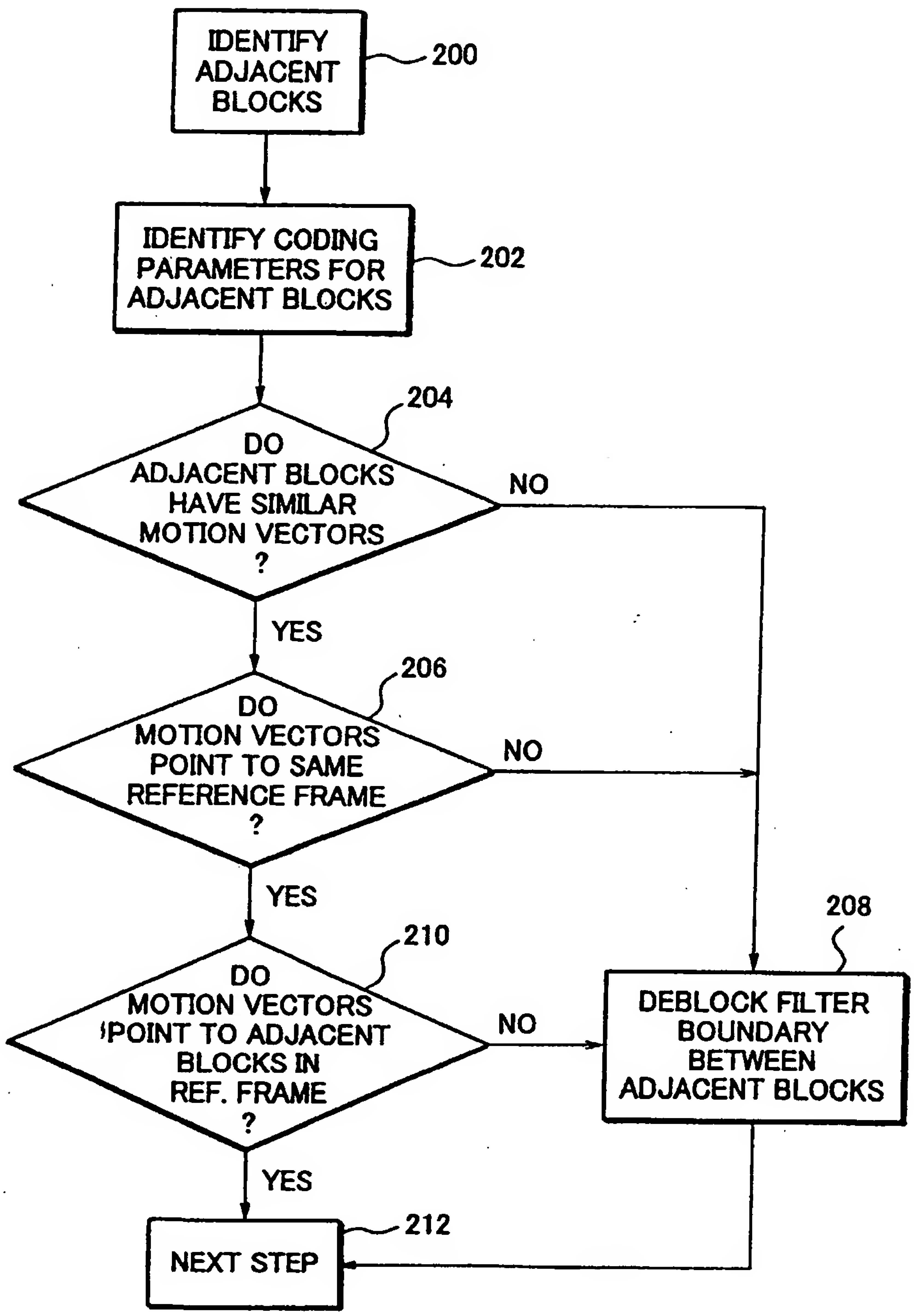

IDENTIFY ADJACENT BLOCKS
200
IDENTIFY CODING PARAMETERS FOR ADJACENT BLOCKS
202
DO ADJACENT BLOCKS HAVE SIMILAR MOTION VECTORS ?
204
NO
YES
DO MOTION VECTORS POINT TO SAME REFERENCE FRAME ?
206
NO
YES
208
DO MOTION VECTORS POINT TO ADJACENT BLOCKS IN REF. FRAME ?
210
NO
DEBLOCK FILTER BOUNDARY BETWEEN ADJACENT BLOCKS
YES
NEXT STEP
212

FIG.12

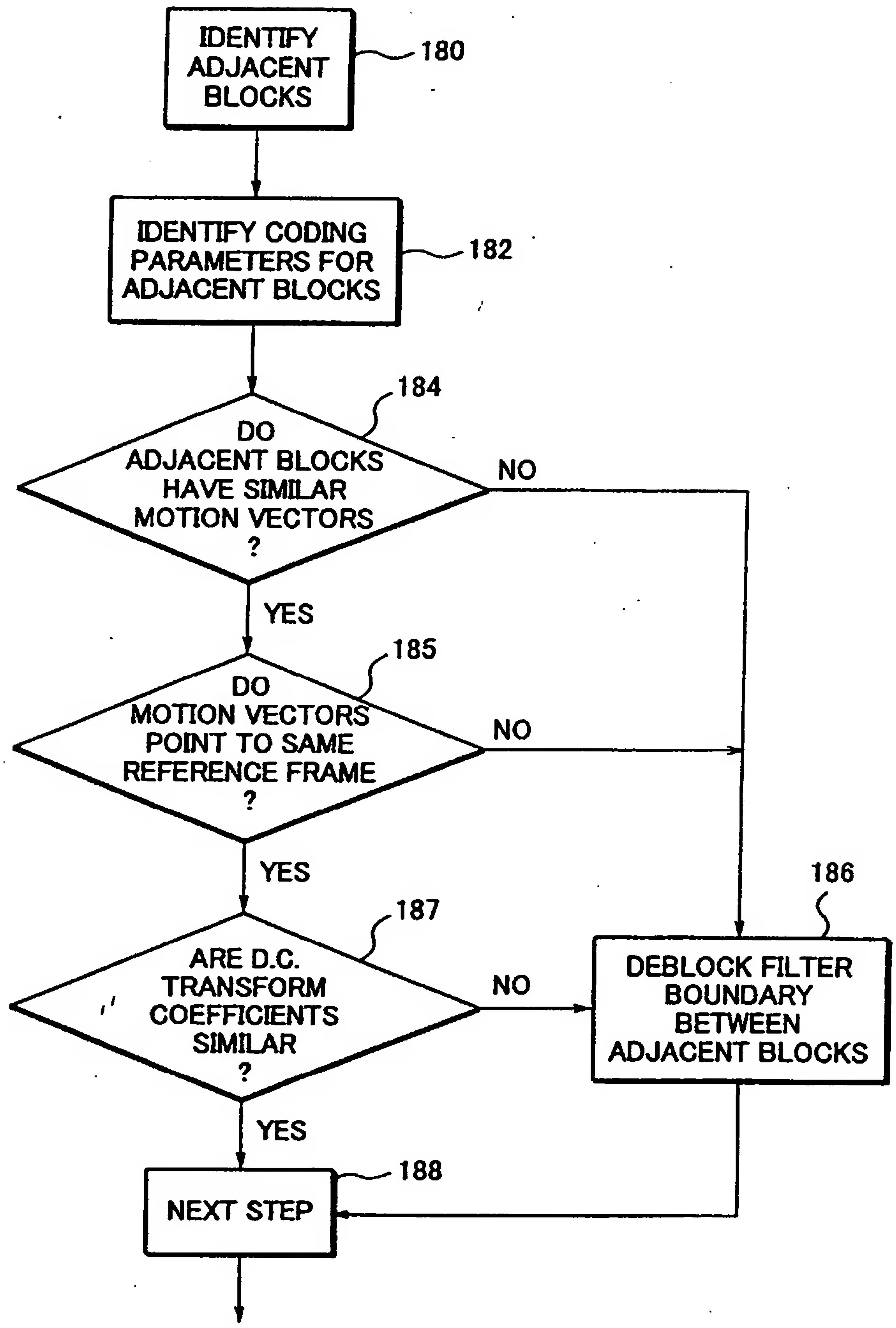

IDENTIFY ADJACENT BLOCKS
180
IDENTIFY CODING PARAMETERS FOR ADJACENT BLOCKS
182
184
DO ADJACENT BLOCKS HAVE SIMILAR MOTION VECTORS ?
NO
YES
185
DO MOTION VECTORS POINT TO SAME REFERENCE FRAME ?
NO
YES
186
187
ARE D.C. TRANSFORM COEFFICIENTS SIMILAR ?
NO
DEBLOCK FILTER BOUNDARY BETWEEN ADJACENT BLOCKS
YES
188
NEXT STEP

FIG.13

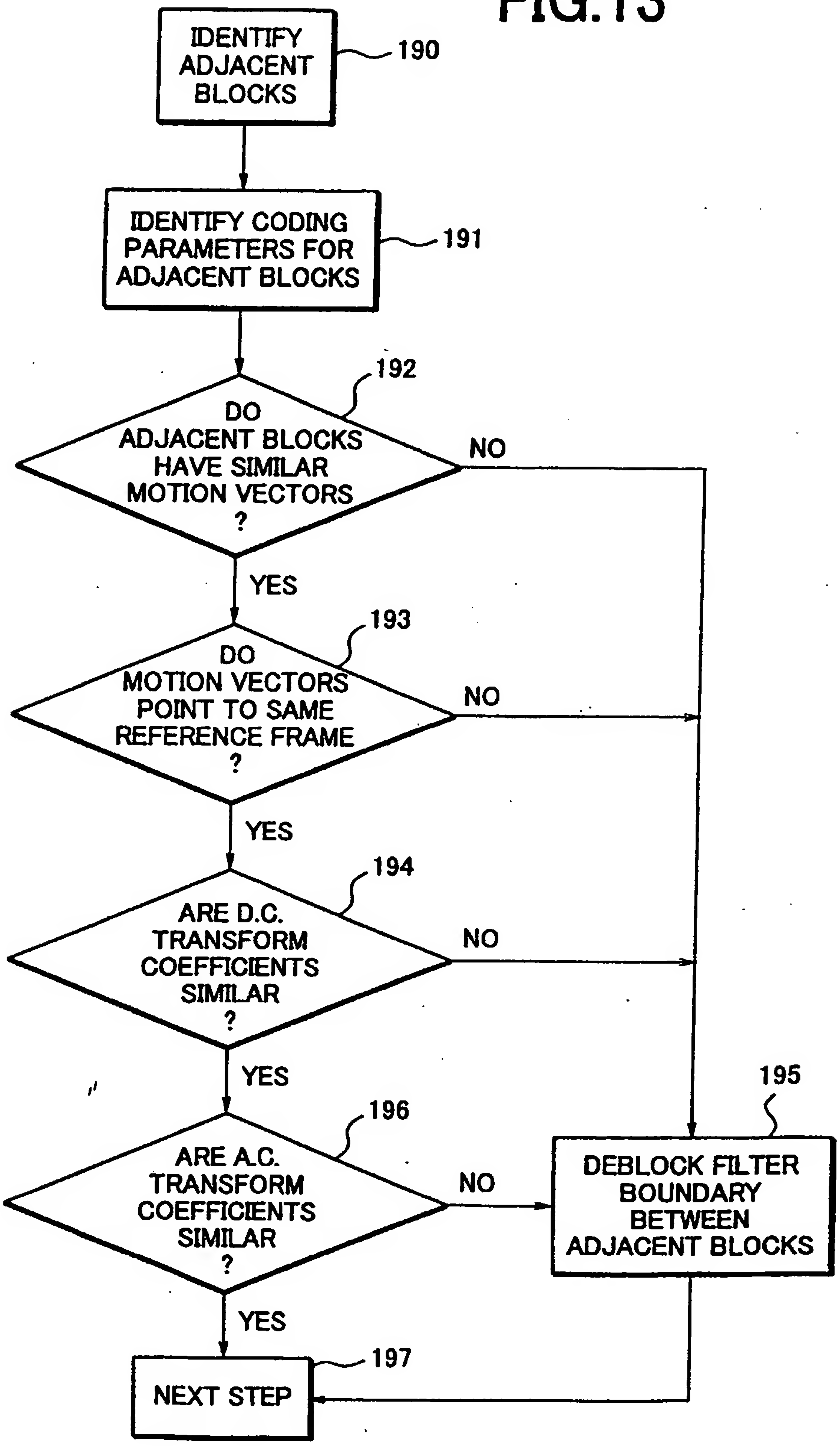

IDENTIFY ADJACENT BLOCKS
190
IDENTIFY CODING PARAMETERS FOR ADJACENT BLOCKS
191
192
DO ADJACENT BLOCKS HAVE SIMILAR MOTION VECTORS ?
NO
YES
193
DO MOTION VECTORS POINT TO SAME REFERENCE FRAME ?
NO
YES
194
ARE D.C. TRANSFORM COEFFICIENTS SIMILAR ?
NO
YES
195
196
ARE A.C. TRANSFORM COEFFICIENTS SIMILAR ?
NO
DEBLOCK FILTER BOUNDARY BETWEEN ADJACENT BLOCKS
YES
197
NEXT STEP

FIG.14

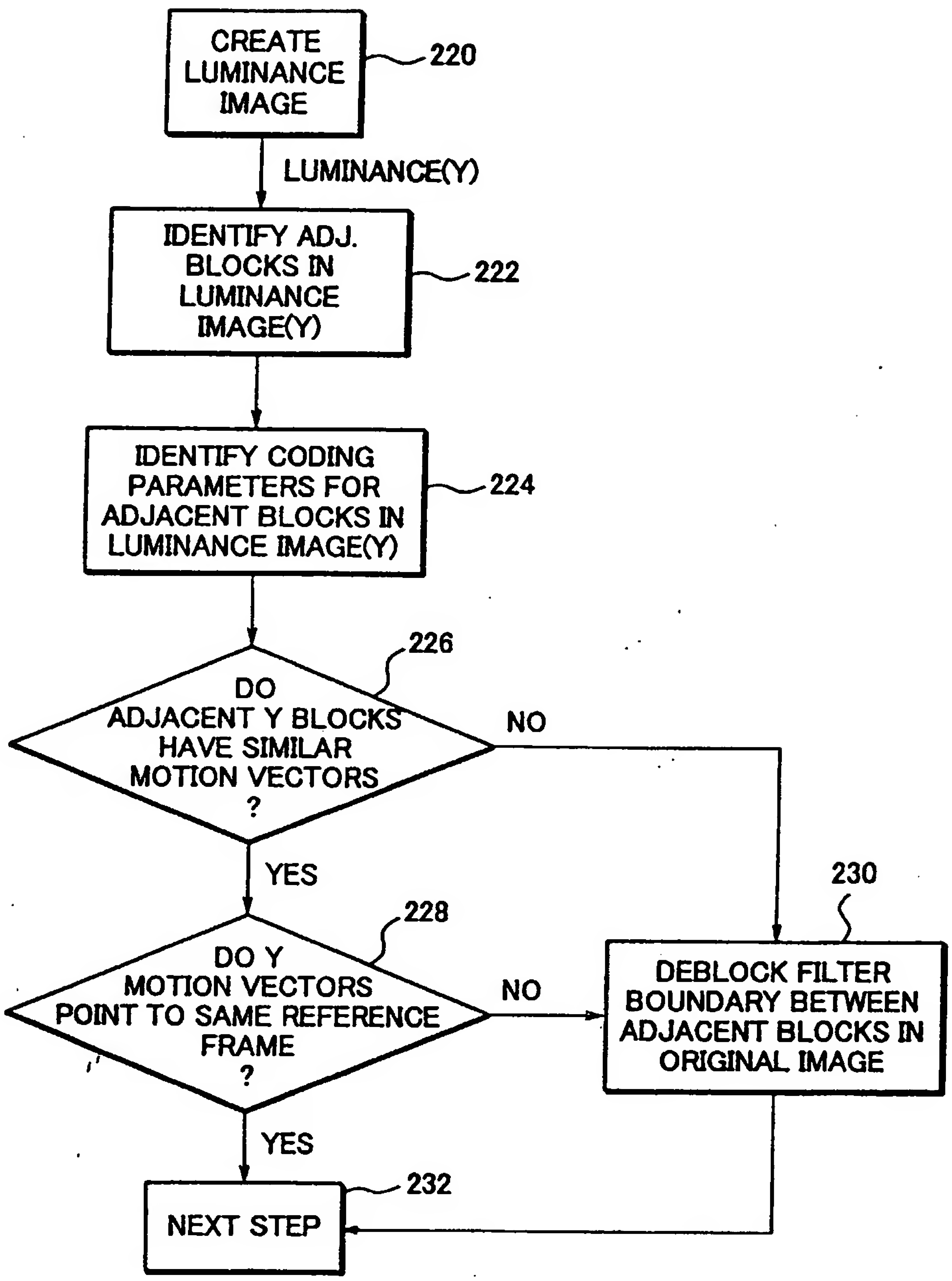

CREATE LUMINANCE IMAGE
220
LUMINANCE(Y)
IDENTIFY ADJ. BLOCKS IN LUMINANCE IMAGE(Y)
222
IDENTIFY CODING PARAMETERS FOR ADJACENT BLOCKS IN LUMINANCE IMAGE(Y)
224
226
DO ADJACENT Y BLOCKS HAVE SIMILAR MOTION VECTORS ?
NO
YES
228
DO Y MOTION VECTORS POINT TO SAME REFERENCE FRAME ?
NO
230
DEBLOCK FILTER BOUNDARY BETWEEN ADJACENT BLOCKS IN ORIGINAL IMAGE
YES
232
NEXT STEP

FIG.15

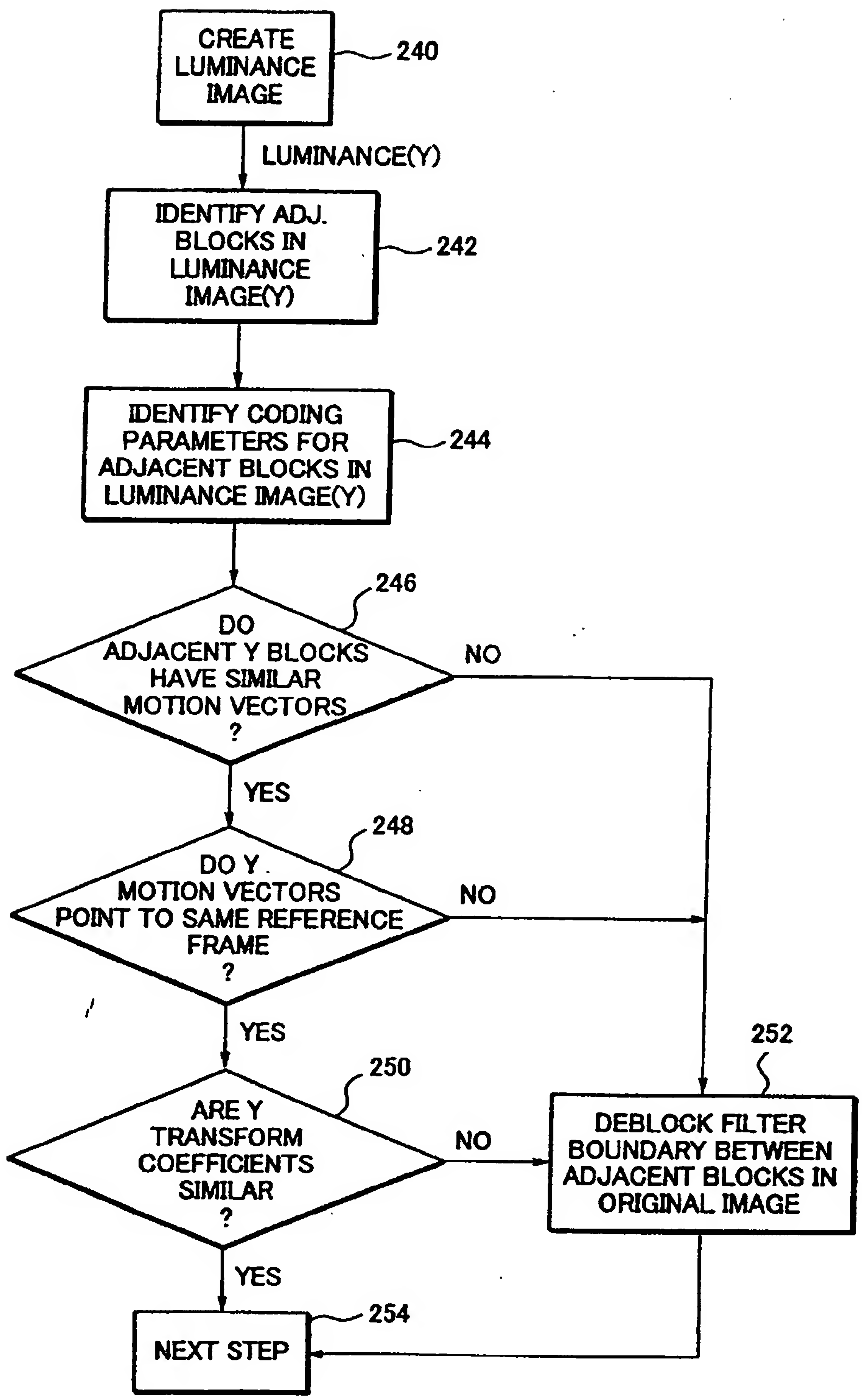

CREATE LUMINANCE IMAGE
240
LUMINANCE(Y)
IDENTIFY ADJ. BLOCKS IN LUMINANCE IMAGE(Y)
242
IDENTIFY CODING PARAMETERS FOR ADJACENT BLOCKS IN LUMINANCE IMAGE(Y)
244
246
DO ADJACENT Y BLOCKS HAVE SIMILAR MOTION VECTORS ?
NO
YES
248
DO Y MOTION VECTORS POINT TO SAME REFERENCE FRAME ?
NO
YES
252
250
ARE Y TRANSFORM COEFFICIENTS SIMILAR ?
NO
DEBLOCK FILTER BOUNDARY BETWEEN ADJACENT BLOCKS IN ORIGINAL IMAGE
YES
254
NEXT STEP

FIG.16

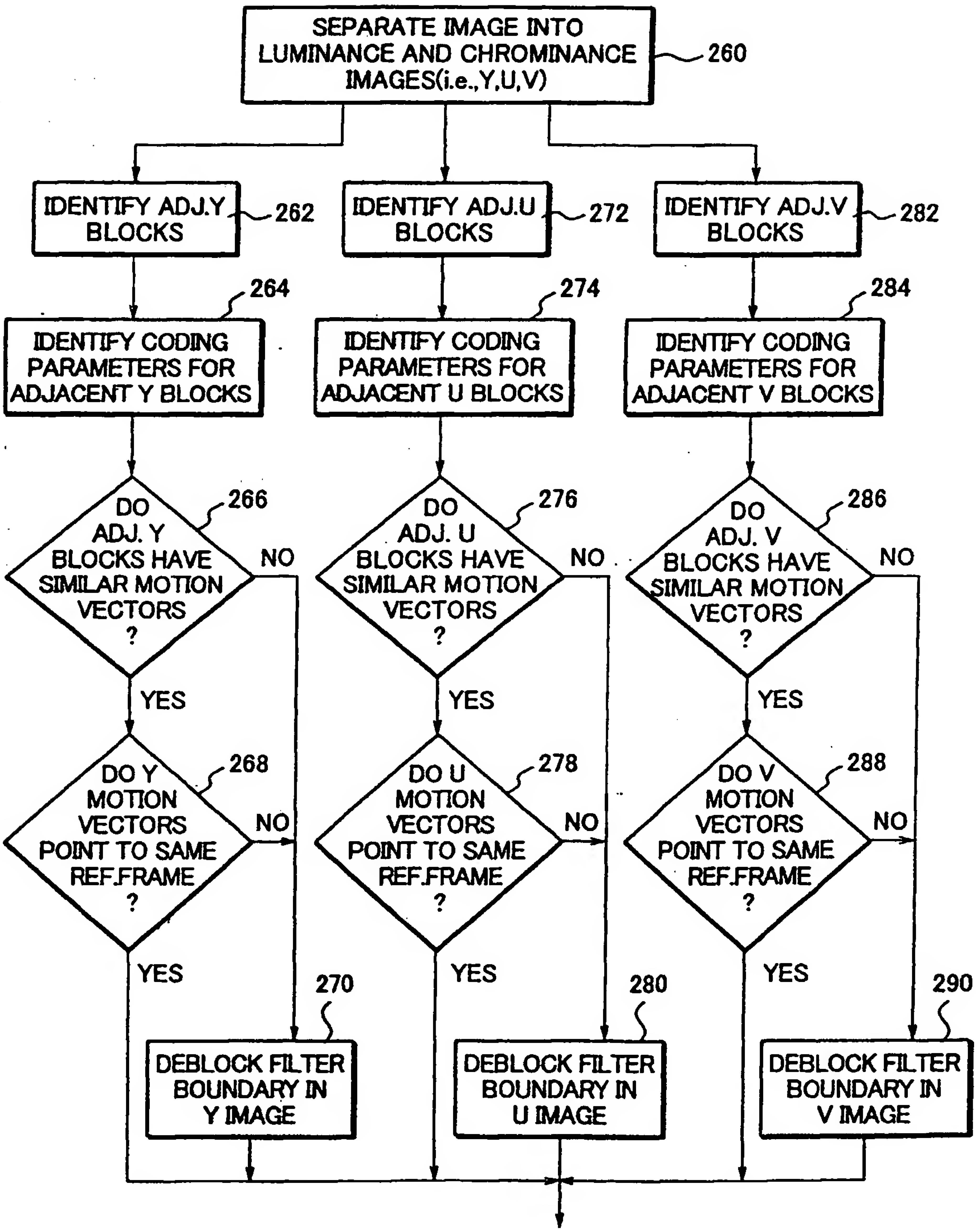
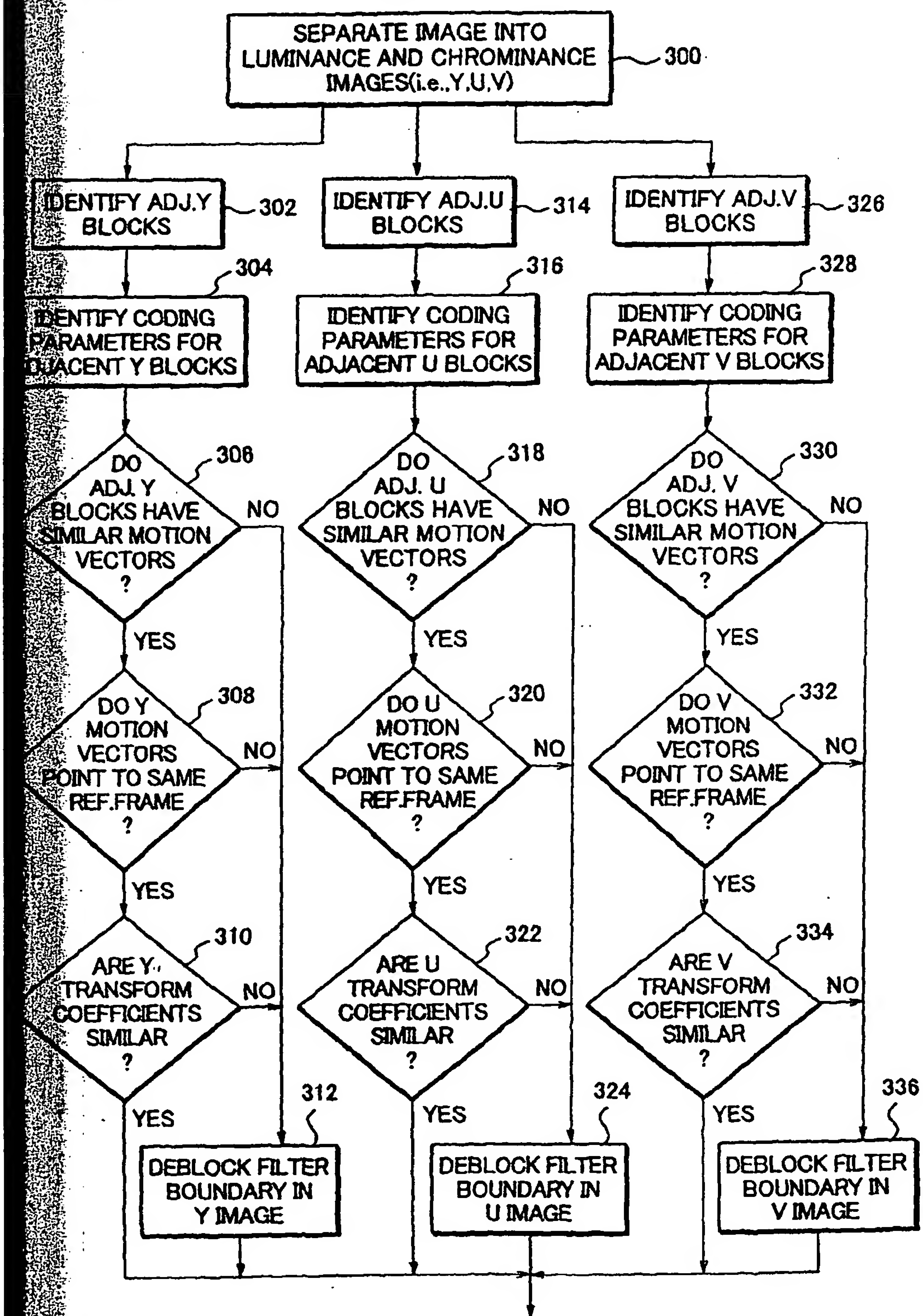

SEPARATE IMAGE INTO LUMINANCE AND CHROMINANCE IMAGES(i.e.,Y,U,V)
260
IDENTIFY ADJ.Y BLOCKS
262
IDENTIFY ADJ.U BLOCKS
272
IDENTIFY ADJ.V BLOCKS
282
264
IDENTIFY CODING PARAMETERS FOR ADJACENT Y BLOCKS
274
IDENTIFY CODING PARAMETERS FOR ADJACENT U BLOCKS
284
IDENTIFY CODING PARAMETERS FOR ADJACENT V BLOCKS
DO ADJ. Y BLOCKS HAVE SIMILAR MOTION VECTORS ?
266
NO
DO ADJ. U BLOCKS HAVE SIMILAR MOTION VECTORS ?
276
NO
DO ADJ. V BLOCKS HAVE SIMILAR MOTION VECTORS ?
286
NO
YES
YES
YES
DO Y MOTION VECTORS POINT TO SAME REF.FRAME ?
268
NO
DO U MOTION VECTORS POINT TO SAME REF.FRAME ?
278
NO
DO V MOTION VECTORS POINT TO SAME REF.FRAME ?
288
NO
YES
YES
YES
270
280
290
DEBLOCK FILTER BOUNDARY IN Y IMAGE
DEBLOCK FILTER BOUNDARY IN U IMAGE
DEBLOCK FILTER BOUNDARY IN V IMAGE

FIG.17


SEPARATE IMAGE INTO LUMINANCE AND CHROMINANCE IMAGES(i.e.,Y,U,V)
300
IDENTIFY ADJ.Y BLOCKS
302
IDENTIFY ADJ.U BLOCKS
314
IDENTIFY ADJ.V BLOCKS
326
304
IDENTIFY CODING PARAMETERS FOR ADJACENT Y BLOCKS
316
IDENTIFY CODING PARAMETERS FOR ADJACENT U BLOCKS
328
IDENTIFY CODING PARAMETERS FOR ADJACENT V BLOCKS
306
DO ADJ. Y BLOCKS HAVE SIMILAR MOTION VECTORS ?
318
DO ADJ. U BLOCKS HAVE SIMILAR MOTION VECTORS ?
330
DO ADJ. V BLOCKS HAVE SIMILAR MOTION VECTORS ?
NO
YES
308
DO Y MOTION VECTORS POINT TO SAME REF.FRAME ?
320
DO U MOTION VECTORS POINT TO SAME REF.FRAME ?
332
DO V MOTION VECTORS POINT TO SAME REF.FRAME ?
310
ARE Y TRANSFORM COEFFICIENTS SIMILAR ?
322
ARE U TRANSFORM COEFFICIENTS SIMILAR ?
334
ARE V TRANSFORM COEFFICIENTS SIMILAR ?
312
DEBLOCK FILTER BOUNDARY IN Y IMAGE
324
DEBLOCK FILTER BOUNDARY IN U IMAGE
336
DEBLOCK FILTER BOUNDARY IN V IMAGE